Gonzalez *et al. Journal of Cloud Computing: Advances, Systems and Applications* (2017) 6:13
DOI 10.1186/s13677-017-0081-4

Journal of Cloud Computing:
Advances, Systems and Applications

REVIEW Open Access

Cloud resource management: towards efficient execution of large-scale scientific applications and workflows on complex infrastructures

Nelson Mimura Gonzalez[1*], Tereza Cristina Melo de Brito Carvalho[1] and Charles Christian Miers[2]

Abstract

Cloud computing evolved from the concept of utility computing, which is defined as the provision of computational and storage resources as a metered service. Another key characteristic of cloud computing is multitenancy, which enables resource and cost sharing among a large pool of users. Characteristics such as multitenancy and elasticity perfectly fit the requirements of modern data-intensive research and scientific endeavors. In parallel, as science relies on the analysis of very large data sets, data management and processing must be performed in a scalable and automated way. Workflows have emerged as a way to formalize and structure data analysis, thus becoming an increasingly popular paradigm for scientists to handle complex scientific processes. One of the key enablers of this conjunction of cloud computing and scientific workflows is resource management. However, several issues related to data-intensive loads, complex infrastructures such as hybrid and multicloud environments to support large-scale execution of workflows, performance fluctuations, and reliability, pose as challenges to truly position clouds as viable high-performance infrastructures for scientific computing. This paper presents a survey on cloud resource management that provides an extensive study of the field. A taxonomy is proposed to analyze the selected works and the analysis ultimately leads to the definition of gaps and future challenges to be addressed by research and development.

Keywords: Cloud computing, Resource management, Scheduling, Workflows

Introduction

Cloud computing evolved from the concept of utility computing, which is defined as the provision of computational and storage resources as a metered service, similar to traditional public utility companies [92]. This concept reflects the fact that modern information technology environments require the means to dynamically increase capacity or add capabilities while minimizing the requirement of investing money and time in the purchase of new infrastructure.

Another key characteristic of cloud computing is multitenancy, which enables resource and cost sharing among a large pool of users [91]. This leads to the centralization of the infrastructure and consequent reduction of costs due to economies of scale [123]. Moreover, the consolidation of resources leads to an increased peak-load capacity as each customer has access to a much larger pool of resources (although shared) compared to a local cluster of machines. Resources are more efficiently used, especially considering that in a local setup they often are underutilized [45]. In addition, multitenancy enables dynamic allocation of these resources which are monitored by the service provider.

Characteristics such as multitenancy and elasticity perfectly fit the requirements of modern data intensive research and scientific endeavors [28]. These requirements are associated to the continuously increasing power of computing and storage resources that in many cases are required on-demand for specific phases of an experiment, therefore demanding elastic scaling. This motivates

*Correspondence: nmimura@larc.usp.br
[1]Escola Politécnica, University of São Paulo (USP), São Paulo/SP, Brazil
Full list of author information is available at the end of the article

the utilization of clouds by scientific researchers as an alternative to using in-house resources [22].

In parallel, as science becomes more complex and relies on the analysis of very large data sets, data management and processing must be performed in a scalable and automated way. Workflows have emerged as a way to formalize and structure data analysis, execute the required computations using distributed resources, collect information about the derived data products, and repeat the analysis if necessary [115]. Workflows enable the definition and sharing of analysis and results within scientific collaborations. In this sense, scientific workflows have become an increasingly popular paradigm for scientists to handle complex scientific processes [150], enabling and accelerating scientific progress and discoveries.

Scientific workflows, like other computer applications, can benefit from virtually unlimited resources with minimal investment. With such advantages, workflow scheduling research has thus shifted to workflow execution in the cloud [111], providing a paradigm-shifting utility-oriented computing environment with unprecedented size of data center resource pools and on-demand resource provisioning [150], enabling scientific workflow solutions to address petascale problems.

One of the key enablers of this conjunction of cloud computing and scientific workflows is resource management [6], which includes resource provisioning, allocation, and scheduling [72]. Even small provisioning inefficiencies, such as failure to meet workflow dependencies on time or selecting the wrong resources for a task, can result in significant monetary costs [22, 135]. Provisioning the right amount of storage and compute resources leads to decisive cost reduction with no substantial impact on application performance.

Consequently, cloud resource management for workflow execution is a topic of broad and current interest [127]. Moreover, there are few researches on scheduling workflows on real cloud environments, and much fewer cloud workflow management systems, which require even further academic study and industrial practice [127]. Workflow scheduling for commercial multicloud environments, for instance, still is an open issue to be addressed [32]. In addition, data transfer between tasks is not directly considered in most existing studies, thus being assumed as part of task execution. However, this is not the case for data-intensive applications [127], especially ones from the big data era, wherein data movement can dominate both the execution time and cost.

Objectives and contributions

This paper surveys over 110 publications on cloud resource management solutions including resource provisioning and task scheduling. The publications were selected from conferences and journals using a systematic search methodology. Our contributions include the definition of a taxonomy used to classify and analyze the publications. The taxonomy was created based on the typical aspects covered by cloud resource management solutions, such as makespan and cost, as well as on aspects pointed by existing works as future challenges for the area, such as reliability and data-intensive loads. Our analysis shows that little to no work is found for specific areas, such as security and dynamic allocation of resources, especially when combined to other aspects such as complex infrastructures and workflow execution. Finally, applying the proposed taxonomy to the publications selected we provide a quantitative assessment of existing solutions, highlighting the future challenges for the execution of large-scale applications on cloud infrastructures.

Document organization

This paper is organized in five main sections. First section, Concepts and definitions, presents the concepts related to cloud resource management, including several definitions and their consolidation. Second section, Resource management taxonomy, presents the taxonomy created to analyze the references selected for the survey. Third section, Survey, presents the results of the survey, including further analysis of specific works to identify gaps and challenges in the field Fourth section, Gaps and challenges, presents the gaps and challenges to be addressed by future research. Fifth and last section, Conclusion, presents the conclusion of this work focusing on the four main problems to be solved in cloud computing resource management.

Concepts and definitions

Cloud computing is a model for enabling on-demand self-service network access to a shared pool of elastic configurable computing resources [76]. The model is driven by economies of scale to reduce costs for users [36] and to allow offering resources in a pay-as-you-go manner, thus embodying the concept of utility computing [7, 8].

In its inception, cloud computing revolved around virtualization as main resource compartmentalization or consolidation strategy [63, 85] to support application isolation and platform customization to suit user needs [17, 18], as well as to enable pooling and dynamically assigning computing resources from clusters of servers [147]. The significant performance improvement and overhead reduction of virtualization technology [81] propelled its adoption as key delivery technology in cloud computing [24]. Nevertheless, developments on Linux Containers and associated technologies [34, 77] led to the implementation of cloud platforms using lightweight containers [44] such as Docker [66, 110] with smaller overhead compared to virtual machines as containers only replicate the libraries and binaries of the virtualized application [53].

Resource management in a cloud environment is a challenging problem due to the scale of modern data centers, the heterogeneity of resource types, the interdependency between these resources, the variability and unpredictability of the load, and the range of objectives of different actors in a cloud ecosystem [52]. Moreover, resource management comprises different stages or resources and workloads. Due to its importance as fundamental building block for cloud computing, several definitions and concepts are found in the literature. The next subsections explore these definitions and provide a consolidated view of cloud resource management.

Singh and Chana

For [108] resource management in cloud comprises three functions: resource provisioning, resource scheduling, and resource monitoring.

Resource provisioning is defined by the authors as the stage to identify the adequate resources for a particular workload based on quality of service (QoS) requirements defined by cloud consumers. This stage includes the discovery of resources and also their selection for executing a workload. The provisioning of appropriate resources to cloud workloads depends on the QoS requirements of cloud applications [21]. In this sense, the cloud consumer interacts with the cloud via a cloud portal and submits the QoS requirements of the workload after authentication. The Resource Information Centre (RIC) contains the information about all the resources in the resource pool and obtains the result based on requirement of workload as specified by user. The user requirements and the information provided by the RIC are used by the Resource Provisioning Agent (RPA) to check the available resources. After provisioning of resources the workloads are submitted to the resource scheduler. Finally, the Workload Resource Manager (WRM) sends the provisioning results (resource information) to the RPA, which forwards these results to the cloud user.

Resource scheduling is defined as the mapping, allocation, and execution of workloads based on the resources selected in the resource provisioning phase [109]. Mapping workloads refers to selecting the appropriate resources based on the QoS requirements as specified by user in terms of SLA to minimize cost and execution time, for instance. The process of finding the list of available resources is referred to as resource detection, while the resource selection is the process of choosing the best resource from list generated by resource detection based on SLA.

Resource monitoring is a complementary phase to achieve better performance optimization. In terms of service level agreements (SLA) both parties (provider and consumer) must specify the possible deviations to achieve appropriate quality attributes. For successful execution of a workload the observed deviation must be less than the defined thresholds. In this sense, resource monitoring is used to take care of important QoS requirements like security, availability, and performance. The monitoring steps include checking the workload status and verifying if the amount of required resources (RR) is larger than the amount of provided resources (PR). Depending on the result more resources are demanded by the scheduler. On the other hand, based on this result the resources can also be released, freeing them for other allocations. Consequently, the monitoring phase also controls the rescheduling activities.

Jennings and Stadler

For [52] resource management is the process of allocating computing, storage, networking and energy resources to a set of applications in order to meet performance objectives and requirements of the infrastructure providers and the cloud users. On one hand, the objectives of the providers are related to efficient and effective resource utilization within the constraints of SLAs. The authors claim that efficient resource use is typically achieved through virtualization technologies, facilitating the multiplexing of resources across customers. On the other hand, the objectives of cloud users tend to focus on application performance, their availability, as well as the cost-effective scaling of available resources based on application demands.

The cloud provider is responsible for monitoring the utilization of compute, networking, storage, and power resources, as well as for controlling this utilization via global and local scheduling processes. In parallel, the cloud user monitors and controls the deployment of its applications on the virtual infrastructure. Cloud providers can dynamically alter the prices charged for leasing the infrastructure while cloud users can alter the costs by changing application parameters and usage levels. However, the cloud user has limited responsibility for resource management, being constrained to generating workload requests and controlling where and when workloads are placed.

The authors distinguish the roles of cloud user from end user. The end user generates the workloads that are processed using cloud resources. The cloud user actively interacts with the cloud infrastructure to host applications for end users. In this sense, the cloud user acts a broker, thus being responsible for meeting the SLAs specified by the end user. Moreover, the cloud user is mostly interested in meeting these requirements in a manner to minimize its own costs of leasing the cloud infrastructure (from the cloud provider) while maximizing its profits.

From a functional perspective, the end user initiates the process by providing one or more workload requests to the workload scheduling component. The requests are relayed to the workload management component provided by the cloud user (broker). The application is submitted to a profiling process that dynamically defines the pricing characteristics, also defining the metrics to be monitored during execution and the objectives (SLAs) to be observed. The cloud user defines the provisioning to be obtained from the cloud provider. The provider receives the requests via a global provisioning and scheduling component that also profiles the requests in order to determine the pricing attributes (this time from cloud provider to cloud user). Moreover, the application is characterized in order to obtain monitoring metrics and objectives from the cloud provider point of view. Finally, the global provisioning and scheduling element submits requests for the local handler, estimating the resource utilization and executing the workloads.

Manvi and Shyam

For [72] resource management comprises nine components:

- *Provisioning*: Assignment of resources to a workload.
- *Allocation*: Distribution of resources among competing workloads.
- *Adaptation*: Ability to dynamically adjust resources to fulfill workload requirements.
- *Mapping*: Correspondence between resources required by the workload and resources provided by the cloud infrastructure.
- *Modeling*: Framework that helps to predict the resource requirements of a workload by representing the most important attributes of resource management, such as states, transitions, inputs, and outputs within a given environment.
- *Estimation*: Guess of the actual resources required for executing a workload.
- *Discovery*: Identification of a list of resources that are available for workload execution.
- *Brokering*: Negotiation of resources through an agent to ensure their availability at the right time to execute the workload.
- *Scheduling*: A timetable of events and resources, determining when a workload should start or end depending on its duration, predecessor activities, predecessor relationships, and resources allocated.

The authors did not explicitly defined the roles or actors related to cloud management activities. The implicit roles in this sense are the cloud provider (responsible for managing the cloud infrastructure) and the cloud user (interested in executing one or more workloads on the cloud infrastructure). QoS is regarded as fundamental part of the resource management premises. In contrast, the SLAs are not explicitly defined as building block for resource management tasks.

Other definitions

For [80], resource management is a process that deals with the procurement and release of resources. Moreover, resource management provides performance isolation and efficient use of underlying hardware. The authors state that the main research challenges and metrics of resource management are energy efficiency, SLA violations, load balancing, network load, profit maximization, hybrid clouds, and mobile cloud computing. No specific remark to cloud roles or to quality of service are made, although the solutions covered by the survey might present QoS related aspects.

For [75], resource management is a core function of cloud computing that affects three aspects: performance, functionality, and cost. In this sense, cloud resource management requires complex policies and decisions for multi-objective optimization. These policies are organized in five classes: admission control, capacity allocation, load balancing, energy optimization, and quality of service guarantees. The admission policies prevent the system from accepting workloads in violation of high-level system policies (e.g., a workload that might prevent others from completing). Capacity allocation comprises the allocation of resources for individual instances. Load balancing and energy optimization can be done either locally or globally, and both are correlated to cost. Finally, quality of service is related to addressing requirements and objectives concerning users and providers. SLA aspects are not explicitly considered in this set of policies.

For [125], resource management is related to predicting the amount of resources that best suits each workload, enabling cloud providers to consolidate workloads while maintaining SLAs.

For [69], resource management comprises two main activities: matching, which is the process of assigning a job to a particular resource; and scheduling, which is the process of determining the order in which jobs assigned to a particular resource should be executed.

Intercorrelation and consolidation

Table 1 presents a summary of the resource management definitions. The table presents the works analyzed for the study of definitions of resource management, a summary of the viewpoints from each work, which are the actors identified in each work, and whether aspects related to Quality of Service and Service Level Agreements are mentioned and considered in the works or not. The importance of identifying these aspects is to analyze the similarities and disparities among the works to allow a better understanding of the definitions.

Some works treat resource management and resource scheduling as the same concept. For instance, [127] present a survey focusing on resource scheduling that also

Table 1 Summary of resource management definitions, actors, and QoS/SLA aspects considered in each definition

Work	Summary	Actors	QoS	SLA
[108]	Three phases: provisioning, scheduling, monitoring	Cloud provider (infra. and workload mgmt.) and cloud consumer (end user)	Yes	Yes
[52]	Organized in three tiers (one per role) and a total of 15 different stages, including scheduling, provisioning, pricing, profiling, and monitoring.	Cloud provider (infra. mgmt.), cloud user (broker), end user (execute workload)	Yes	Yes
[72]	Nine components: provisioning, allocation, adaptation, mapping, modeling, estimation, discovery, brokering, and scheduling	No specific actors are identified; implicit assumption of at least two roles: cloud provider and cloud user	Yes	No
[80]	Two tasks: procurement and release of resources. Two objectives: performance isolation and efficient use of hardware. Seven metrics: energy, SLA, load, network load, profit, hybrid clouds, and mobile clouds.	No specific actors are identified; implicit assumption of at least two roles: cloud provider and cloud user	No	Yes
[75]	Three aspects affected: performance, functionality, and cost. Five policy classes: admission control, capacity allocation, load balancing, energy optimization, and QoS.	Cloud provider and cloud user.	Yes	No
[125]	Predicting workload to enable workload consolidation while meeting SLAs.	No specific actors	Yes	Yes
[69]	Two main activities: matching and scheduling.	No specific actors	Yes	Yes

comprises several of the components proposed by [72], such as provisioning, allocation, and modeling.

Three definitions were selected due to their clear definition of steps and components of resource management. Table 2 provides a summary of the phases or steps proposed by each definition.

While the definition from [72] proposes more steps than the others, there is a natural correlation between the phases proposed by each definition. Table 3 presents the correlation between the phases from [108] and the other two. The objective of this table is to fit the steps proposed by [52] and by [72] into the steps from [108], which represents a simpler classification of resource management tasks.

Comparing [52] to [108], the workload profiling (to assess the resource demands), pricing, and provisioning steps defined by [52] fit the provisioning step from [108], which is essentially the phase to identify the resources for a particular workload based on its characteristics and on the QoS. This includes the selection of resources to execute the workload. These aspects fit the steps of discovery, modeling, brokering, and provisioning from [72]. Note that the brokering aspect is also implicitly included in the definition from Jennings and Stadler, as they define a specific role for the brokering activity (the cloud user; the end user is the actor that has a workload to be executed in the cloud).

The scheduling phase from [108] are organized in estimation and scheduling by [52]. Manvi and Shyam [72] include an allocation step to these two. In summary, these steps represent the mapping, allocation, and execution of the workload based on the resources selected in the provisioning phase.

Finally, the monitoring phase is present in [108] and in [52]. For [72] the monitoring tasks are implicitly included by the adaptation step, which is related to dynamically adjusting resources to fulfill workload requirements. Because it is necessary to monitor both resource availability and workload conditions in order to provide this feature, this means that this step directly relies on some form of monitoring.

In terms of consolidation, the common point of all definitions is the aspect of managing the life cycle of resources

Table 2 Explicit phases or steps proposed in each definition

Work	Explicit phases or steps
[108]	Provisioning, Scheduling, Monitoring
[52]	Profiling, Pricing, Provisioning, Estimation, Scheduling, Monitoring
[72]	Provisioning, Allocation, Adaptation, Mapping, Modeling, Estimation, Discovery, Brokering, Scheduling

Table 3 Correlation between steps defined in [52] and [72] compared to [108]

[108]	[52]	[72]
Provisioning	Profiling, Pricing, Provisioning	Discovery, Modeling, Brokering, Provisioning
Scheduling	Estimation, Scheduling	Estimation, Mapping, Allocation, Scheduling
Monitoring	Monitoring	Adaptation

and their association to the execution of tasks. This is the central governing point of cloud resource management which is independent of a specific phase of this life cycle. While it is fundamental to distinguish each phase, they all contribute to two ultimate purposes:

- Enable task execution; and
- Optimize infrastructural efficiency based on a set of specified objectives.

These are the key points of interest of this work, therefore comprising not only the specific task of scheduling resources (i.e., associating them to a task), but also managing the resource from its initial preparation (e.g., discovery) to its utilization and distribution.

Resource management taxonomy

Because of its relevance, cloud computing resource management is a topic that not only has a lot of work and research, but also existing surveys and taxonomies. This section presents an analysis of existing taxonomies used to classify the resource management solutions. Finally, we present the taxonomy proposed for classifying the works analyzed in this survey.

Relevant work

Bala and Chana [9] definee nine categories to classify resource management and scheduling solutions: time, cost, scalability, scheduling success rate, makespan, speed, resource utilization, reliability, and availability. Among these categories, time, speed, and makespan are directly correlated. Resource utilization is related to the efficiency of utilization of resources, which is a fundamental aspect of any algorithm. Reliability and availability aspects, although defined as categories, were not identified in any of the solutions analyzed by the authors.

Sotiriadis et al. [112] classify the solutions in terms of flexibility, scalability, interoperability, heterogeneity, local autonomy, load balancing, information exposing, real-time data, scheduling history records, unpredictability management, geographical distribution, SLA compatibility, rescheduling, and intercloud compatibility. Several properties are relevant for heterogeneous environments, such as local autonomy and geographical distribution. Others are correlated, such as scalability, unpredictability management, and rescheduling.

Wu et al. [127] use nine categories to classify their references:

- *Best-effort*: Optimize one objective while ignoring other factors such as QoS requirements.
- *Deadline-constrained*: Scheduling based on the trade-off between execution time and monetary cost under a deadline constraint.
- *Budget-constrained*: The objective is to finish a workflow as fast as possible at given budget.
- *Multi-criteria*: Several objectives are taken into account.
- *Workflow-as-a-service*: Multiple workflow instances submitted to the resource manager.
- *Robust scheduling*: Able to absorb uncertainties such as performance fluctuation and failure.
- *Hybrid environment*: Able to address requirements of hybrid clouds.
- *Data-intensive*: Data-aware workflow scheduling.
- *Energy-aware*: Able to save energy while optimizing execution.

The authors also mention other properties such as makespan (which fits the Best-Effort category). Moreover, the multi-criteria category represents the convergence of several objective functions, such as cost and performance. Workflow-as-a-Service (WaaS) is the scheduling of multiple workflows onto a cloud infrastructure. Robust scheduling refers both to reliability and to performance fluctuations, both factors that can affect the performance and consequently the effectiveness of a schedule. Finally, hybrid environments, data-intensive workflows, and energy-aware scheduling represent the novel challenges in terms of cloud scheduling resource management according to the authors.

Singh and Chana [108] define a taxonomy based on twelve properties:

- *Cost-based*: Organized in multi-QoS, virtualization-based, application-based, and scalability-based.
- *Time-based*: Organized in deadline-based and combination of deadline and budget.
- *Compromised Cost-Time*: Based either on workflows or workloads.
- *Bargaining-based*: Organized in market-oriented, auction, and negotiation.
- *QoS-based*: Based on several QoS aspects, including security and resource utilization.
- *SLA-based*: Based on several SLA types, including workload and autonomic aspects.
- *Energy-based*: Combined with deadlines and SLAs.
- *Optimization-based*: Optimization of several combinations of parameters.
- *Nature Inspired and Bio-Inspired*: Including genetic algorithms and ant colony approaches.
- *Dynamic*: Several combinations of aspects with dynamic management.
- *Rule-based*: Special cases for failures and hybrid clouds.
- *Adaptive-based*: Prediction-based and Bin-Packing strategies.

Several of the categories have direct correlations, and some are used to combine the aspects covered in other categories, such as optimization-based and the dynamic category.

Proposed taxonomy

The consolidated taxonomy focuses on addressing the requirements of heterogeneous environments composed by multiple environments (e.g., hybrid clouds and multicloud scenarios), with data-intensive workflows and high level of dynamic mechanisms. Also, properties from prior work were selected by identifying the commonalities between the works analyzed and also based on future challenges for large-scale execution of applications and workflows, such as data-intensive workflows, hybrid and multicloud scenarios, performance fluctuation, and reliability.

- *Makespan/Time*: encompasses all aspects related to run time and time-based optimization.
- *Deadline*: encompasses aspects also related to time but associated to predefined limits to finish a workflow – the central idea is not to finish the execution of a workflow as fast as possible, but simply to address a specific deadline and possibly save resources (i.e., reduce resource allocation) as long as the deadline is met.
- *Cost/Budget*: encompasses all aspects related to financial cost and benefits, such as cost minimization and budget limitation.
- *Data-Intensive*: works that effectively encompass one or more aspects inherent to data-intensive workflows.
- *Dynamic*: works that employ some form of dynamic mechanism to continuously adjust the scheduling decision. This is a typical method to address issues related to unpredictability, such as performance fluctuation.
- *Reliability*: works that encompass some form of reliability-related aspect, such as selecting nodes in a way to minimize the chances of failure, or providing mechanisms to circumvent failures.
- *Security*: works that consider any aspect of security (in the sense of confidentiality).
- *Energy*: energy-aware scheduling mechanisms.
- *Hybrid/Multicloud*: works that address requirements of hybrid clouds and multicloud scenarios.
- *Workload/Workflow*: works that address requirements for scheduling workflows on clouds.

Compared to the other taxonomies, the proposed one encompasses some of the fundamental properties connected to the QoS components that govern the scheduling decisions, such as makespan, cost, deadline, energy, etc. These properties are fully or at least partially covered by the other taxonomies, such as [9], with cost, makespan, and reliability; [112], with unpredictability management (closely related to dynamic properties and reliability) and rescheduling; [127], with deadline, budget, reliability, and energy; and [108], with cost, time, and energy. In addition, the proposed taxonomy encompasses some of the attributes of interest to this work, such as hybrid and multicloud aspects, and workflow resource management.

Survey

The method used to identify the surveys and other related work is based on searches performed in the following engines: IEEE Xplore, ACM Digital Library, ScienceDirect, Scopus, and Google Scholar. Moreover, two main search queries were used: “cloud scheduling survey” and “cloud resource management survey” (both without quotes). Some results were immediately discarded, such as ones addressing mobile cloud computing or other specific scenarios, such as Internet of Things and sensor networks. The focus of this analysis is to identify the surveys and taxonomies for cloud computing resource management focusing on five aspects: data-intensive loads, dynamic management, reliability, hybrid/multicloud scenarios, and workflow management. Works that do not cover at least one of these topics were not further analyzed, unless they represent solutions that led to the creation of others that do cover these aspects, such as DCP [57] and HEFT [117]. This led to selection of 113 works related to resource management and task scheduling with the majority focusing on cloud computing and a few works on distributed systems, such as [51] and [105]. The Table 4 shows the works, their highlights (very brief summary of contributions or main aspects addressed), and whether each category of the taxonomy was addressed or not. For each category three levels were considered:

- Fully addressed: The work provides a solution that focuses on addressing the specific aspect, with clear mechanisms to cover it and potentially with experiments showing the effectiveness. For instance, [15] explicitly defines mechanisms to address the requirements of hybrid clouds.
- Partially addressed: The work provides mechanisms that could be used to address the specific aspect, even if not explicitly mentioned in the work. For instance, [42] does not directly address deadline and cost aspects, but the solution proposed could be used to cover them with slight operational modifications.
- Not addressed: The work does not address the aspect.

The majority of the works focus on aspects related to cost and time, such as makespan deadline-based solutions. Among them, makespan is addressed by 44 works (39%), deadlines are addressed by 31 works (27%), and cost is

Table 4 Summary of identified related work classified using the consolidated taxonomy

Work	Highlights	MK	DL	CT	DT	DY	RL	SC	EN	HM	WL
[105]	Dynamic level scheduling (DLS)	x	.	.	o	x	.	.	.	.	x
[126]	Wide-area scheduling with dynamic load balancing	.	.	.	x	x	.	.	.	o	.
[57]	Dynamic Critical Path (DCP)	o	.	.	.	x	.	.	.	.	x
[99]	Integration to conventional schedulers.	.	.	.	.	.	.	.	.	o	.
[2]	ELISA, decentralized dynamic algorithm	.	.	.	.	x	.	.	.	o	.
[51]	Hierarchical scheduling	o	.	.	.	.	.	.	.	.	o
[27]	Federation of resource traders	.	.	.	.	.	.	.	.	o	.
[117]	HEFT (Heterogeneous Earliest Finish Time)	x	.	.	.	.	.	.	.	.	.
[113]	Redundantly distribute job to multiple sites to increase backfilling	.	.	.	.	.	.	.	.	o	.
[30]	Performance and reliability optimization	x	.	x	.	x	x	.	.	.	x
[16]	Reduce maximum job waiting time in the queue	x	.	.	.	.	.	.	.	o	.
[3]	Community of peers for brokering	.	.	.	.	o	.	.	.	o	.
[49]	Fault-tolerant scheduling	.	.	.	.	.	x	.	.	.	x
[79]	Dynamic, deadline, energy	.	x	.	.	x	.	.	x	.	x
[97]	Rescheduling policies	x	.	o	.	x	.	.	.	.	x
[58]	Auction-based scheduling.	.	.	.	.	.	.	.	.	o	.
[138]	Deadline partitioning	.	x	o	.	.	.	.	.	.	x
[121]	Dynamic voltage scaling	.	x	o	.	o	.	.	x	.	x
[137]	Genetic algorithm to optimize cost with deadline constraint	.	x	x	.	.	.	.	.	.	x
[149]	Merge multiple DAGs	x	.	.	.	.	.	.	.	.	x
[104]	Makespan and robustness	x	.	.	.	.	x	.	.	.	x
[102]	Load balancing on arrival	.	.	.	.	o	.	.	.	o	.
[98]	LOSS and GAIN approaches	x	.	x	.	.	.	.	.	.	x
[43]	Performance and reliability optimization	x	.	.	.	.	x	.	.	.	x
[31]	Reliable HEFT	x	.	.	.	.	x	.	.	.	x
[139]	Minimize execution time and cost	x	x	x	.	.	.	.	.	.	x
[71]	Dynamic scheduling	.	.	.	.	x	o	.	.	.	x
[90]	Dynamic storage mgmt.	o	.	.	x	.	.	.	.	.	x
[55]	Energy and deadline	.	x	.	.	o	.	.	x	.	x
[145]	Forecast prototype and SLA compensation	.	.	x	.	.	.	.	.	.	.
[146]	Historical information, forecasting	.	.	x	.	.	.	.	.	.	.
[47]	Delegated matchmaking, local vs remote usage	.	o	.	.	o	.	.	.	o	.
[29]	Improve average response time	o	.	.	.	.	.	.	.	o	.
[142]	Float time amortization	.	x	o	.	.	.	.	.	.	x
[142]	Based on HEFT	x	.	.	.	.	.	.	.	.	x
[83]	Bandwidth speedup, data-intensive	o	.	.	x	.	.	.	.	.	x
[89]	Makespan and energy	x	.	.	.	.	.	.	x	.	x
[133]	MQMW (Multiple QoS scheduling of Multi-Workflows)	x	.	x	.	x	.	.	.	.	x
[84]	RASA (Resource-Aware Scheduling Algorithm	x	.	.	.	.	.	.	.	.	.
[59]	Decentralized model that improves makespan	x	.	.	.	.	.	.	.	o	.
[35]	Fuzzy approach for decentralized grids	o	.	.	.	.	.	.	.	o	.
[93]	Backfilling strategy based on dynamic information	x	.	.	.	x	.	.	.	o	.
[23]	Ant Colony Optimization	x	x	x	.	.	x	.	.	.	x
[140]	Path-based deadline partition	.	x	.	.	.	.	.	.	.	x

Table 4 Summary of identified related work classified using the consolidated taxonomy (*Continued*)

[141]	Greedy time-cost distribution	.	.	x	.	.	.	.	.	.	x
[61]	Optimize makespan and resource utilization	x	.	.	.	x	.	.	.	.	x
[114]	Similar to YU et al., 2007	x	x	x	.	.	.	.	.	.	x
[13]	Data staging	x	o	.	x	.	.	.	.	.	x
[96]	QoS-aware, cost and execution time	x	.	x	.	.	.	.	.	.	.
[153]	Based on genetic algorithm; increase resource utilization	.	.	.	.	x	.	.	.	.	.
[100]	Cost-based	.	.	x	.	.	.	.	.	.	.
[67]	Time-cost-based, instance-intensive workflows	x	x	x	.	.	.	.	.	.	x
[82]	Particle swarm optimization heuristic;	.	.	.	x	x	.	.	.	.	x
[94]	Brokering for multiple grids.	.	.	.	.	.	.	.	.	o	.
[122]	Bidding system for resource selection	o	.	.	.	.	.	.	.	.	.
[128]	PSO to minimize cost with deadline constraint	o	x	x	.	.	.	.	.	.	x
[39]	Optimize makespan and cost	x	.	x	.	.	.	.	.	.	x
[88]	Dynamic programming	x	.	x	.	.	x	.	.	.	x
[25]	Dynamic scheduling	.	.	.	.	x	o	.	.	.	x
[11]	Energy efficiency	.	.	.	.	o	.	.	x	.	.
[131]	Reputation-based QoS provisioning	o	o	x	.	.	.	.	.	.	.
[74]	Deadline, budget, auto-scaling	.	x	x	.	o	.	.	.	.	.
[64]	SHEFT (Scalable HEFT)	x	.	.	x	x	.	.	.	.	x
[119]	OWS (Optimal Workflow Scheduling);	x	.	.	.	.	.	.	.	.	x
[132]	Justice-based scheduling	x	.	.	.	o	.	.	.	o	.
[151]	Budget-constrained HEFT	.	.	x	.	x	.	.	.	.	x
[62]	CCSH to minimize makespan and cost	x	.	x	.	.	.	.	.	.	x
[19]	Deadline optimization based on delaying	.	x	.	.	x	.	.	.	.	x
[73]	Multiple DAGs; deadline-based	o	x	o	.	x	.	.	.	.	x
[15]	Hybrid clouds; iteratively resch. tasks until mksp.; deadline	x	x	o	.	x	.	.	.	x	x
[60]	Makespan and energy	x	.	.	.	.	.	.	x	.	x
[78]	Makespan and energy	x	.	.	.	o	.	.	x	.	x
[116]	MapReduce on public clouds	.	x	x	.	.	.	.	.	.	.
[50]	Multi-tier applications	o	.	o	.	.	.	.	.	.	.
[143]	Auction-based, cloud-provider viewpoint	.	.	.	.	.	.	.	.	.	.
[40]	Heterogeneous workloads	.	x	.	.	.	.	.	.	.	.
[54]	SLA management, improve resource utilization	.	o	o	.	.	.	.	.	.	o
[107]	Multi-cloud, cost optimization	.	.	x	.	.	.	.	.	x	.
[37]	Multi-objective, cost constraints	.	.	x	.	.	.	.	.	x	.
[144]	Backtracking and continuous cost evaluation	o	.	x	.	x	.	.	.	.	x
[33]	Multi-objective scheduling	x	.	x	o	.	x	.	x	.	x
[12]	Pareto-based; execution time and cost	x	.	x	.	.	.	.	.	.	x
[118]	Combination of DAG merging techniques	x	.	.	.	.	.	.	.	.	x
[70]	Auto-scaling of resources	o	x	x	o	.	.	.	.	.	x
[124]	Fault-tolerant scheduling	.	x	x	.	o	x	.	.	.	x
[120]	Deadline-driven, scientific applications, hybrid clouds	.	x	.	.	.	.	.	.	x	o
[148]	Energy-aware, scheduling delay	.	o	.	.	o	.	.	x	.	.
[20]	Aneka platform; QoS-driven, hybrid	.	x	.	.	o	.	.	.	x	.
[48]	Cost minimization, deadline	.	x	x	.	.	.	.	.	.	.
[26]	Negotiation/bargaining	.	x	x	.	.	.	.	.	.	.

Table 4 Summary of identified related work classified using the consolidated taxonomy (*Continued*)

[129]	Market oriented	x	.	x	.	.	.	.	.	.	x
[46]	Community-aware decentralized dynamic scheduling	o	.	.	.	o	.	.	.	o	.
[1]	Partial Critical Path (PCP)	o	x	o	.	.	.	.	.	.	x
[65]	Minimize end-to-end delay	o	.	x	.	.	.	.	.	.	x
[152]	Monte Carlo approach	x	.	o	.	x	.	.	.	.	x
[103]	Power aware scheduling	x	.	.	.	o	.	.	x	.	x
[134]	Particle swarm optimization	x	.	x	.	o	.	.	x	.	x
[130]	Data-intensive, energy-aware	x	.	.	x	o	.	.	x	.	x
[42]	Rule-based	.	o	o	.	.	.	.	.	x	.
[38]	Energy, deadline	.	x	o	.	.	.	.	x	.	.
[41]	Bag of tasks, time and cost	x	.	x	.	.	.	.	.	.	.
[106]	SLA-based cost model; power	o	.	x	.	.	.	.	o	.	.
[136]	Cost management	.	.	o	.	.	.	.	.	.	.
[5]	Predict Earliest Finish Time (PEFT)	x	.	.	o	.	.	.	.	.	x
[14]	Cat Swarm Optimization	.	.	x	x	.	.	.	.	.	x
[95]	PSO considering performance variation and VM boot time	.	x	x	.	x	.	.	.	.	x
[4]	Aggregation-based budget distribution	.	.	x	.	.	.	.	.	.	x
[87]	Critical-path heuristic	x	x	x	.	.	x	.	.	.	x
[86]	Spot instances	.	x	x	.	x	x	.	.	.	x
[10]	Fault-tolerance	.	.	x	.	o	x	.	.	.	.
[56]	Behavioral-based estimation	.	.	o	.	x	.	.	.	.	.
[154]	Multiple workflows, optimize time and cost	x	x	x	.	.	.	.	.	.	x
[68]	Multi-cloud, enhanced workflow model	o	x	x	x	.	.	.	.	x	x

MK = Makespan/Time; DL = Deadline; CT = Cost/Budget; DT = Data-Intensive; DY = Dynamic; RL = Reliability; SC = Security; EN = Energy; HM = Hybrid/Multicloud; WL = Workload/Workflow; . = Not addressed; x = Fully addressed; o = Partially addressed

addressed by 43 works (38%). In contrast, none of solutions address security aspects related to confidentiality, such as safe zones to execute code and to store sensitive data.

Regarding support for workflows and workloads, 64 works (57%) provide some level of support to execute workflows using the resource management solution proposed. However, when combined to aspects related to dynamic placement and replacement of resources and tasks, only 19 (17%) provide support for both aspects (dynamic execution of workflows). Combining workflow support to data-intensive workflows leads to only 8 works (7%). Finally, combining workflow support to hybrid and multicloud scenarios, only 2 works (2%) address both aspects. None of the works combine workflow support, data-intensive loads, hybrid and multicloud scenarios, dynamic scheduling and rescheduling, and reliability aspects.

Data-intensive loads are explicitly supported by only 9 works (8%). Hybrid and multicloud scenarios are supported by 7 works (6%). This analysis reveals that while there are works addressing these aspects in separate, none provide explicit support for all aspects of interest and regarded as challenges for future deployments.

Further analysis

This subsection presents the works that were selected for further analysis to identify gaps and future challenges for cloud resource management regarding the execution of large-scale applications and workflows. The analysis of these works is summarized by Table 5.

Pandey et al. [82] propose a heuristic based on PSO that considers both computation and data transmission costs. The workflow is modeled as a DAG. Transfer cost is calculated according to the bandwidth between sites. Average cost of communication between two resources is considered to be applicable only when two tasks have file dependency between them. For two or more tasks executing on the same resource the communication cost is assumed to be zero. This implies no cost relative to sequential accesses to a file (e.g., the input file), but a rather uniform distribution of content among nodes. On the other hand, for a data-intensive workflow with large inputs and several I/O-heavy intermediary phases, even the cost of accessing resources on the same node cannot be overlooked.

Table 5 Summary of further analysis

Work	Data transfers and imbalance	Dynamic scheduling	Hybrid and Multicloud	Workflow support
PANDEY et al., 2010	Transfers are evaluated via workflow DAG and resource allocation; transfer imbalance is not addressed.	Only addresses fluctuations in the transfer costs. Other aspects such as performance fluctuations and reliability are not mentioned.	No explicit support or experiments.	Modeled as DAGs; richer characterizations are not supported.
LIN; LU, 2011	Transfer capacity of nodes in the same network are assumed to be uniform. Transfer imbalance is discarded.	Not addressed.	No explicit support or experiments	Supported; no details included.
XU et al., 2009	Transfers and data properties are not explicitly addressed.	Not addressed.	No explicit support or experiments.	Multiple workflows supported via common merging point; simple DAG modeling.
WEISSMAN; GRIMSHAW, 1996	Data locality is a scheduling constraint; worker must be assigned closer to data.	Two levels: local and global. Rescheduling is first handled on local level. Details are not provided.	Design for wide-area systems (pre-dates cloud computing).	No explicit support.
CHEN; ZHANG, 2009	Data communication and transfers are not explicitly addressed.	Not addressed.	No explicit support or experiments.	Simplified DAG model without edge costs.
RODRIGUEZ; BUYYA, 2014	Rigidly modeled; fixed costs for transfers and no cost for local I/O.	Not addressed.	Not addressed.	DAG with fixed transfer costs and computation costs based on FLOPS.
FARD et al., 2012	Transfers are considered but contention effects are not. Energy calculations ignore transfer times.	Not addressed.	No explicit support or experiments.	DAG with fixed transfer costs; not details on task costs.
MALAWSKI et al., 2012	Algorithm does not consider the size of input data; transfer time is part of computation.	Initial scheduling plus periodic adjusting depending on amount of idle resources.	No explicit support or experiments.	DAG with fixed transfer costs and computation costs with slight variability.
SAKELLARIOU; ZHAO, 2004	Linear variation to amount of input data size.	Immediately before execution of tasks and bound to a condition to minimize number of reschedules.	Not addressed; solution originally designed for grids.	DAG with computation and transfer costs modeled with linear variation w.r.t. amount of input.
WANG; CHEN, 2012	Not addressed. DAG does not specify transfer costs.	Not addressed.	No explicit support.	DAG with tasks and implicit costs. No transfer costs and no more complex characterization.
POOLA et al., 2014a	Based on data size and one value for network bandwidth.	Not addressed.	No explicit support.	DAG with task cost based on number of instructions.
BITTENCOURT; MADEIRA, 2011	Based on data size and fixed network bandwidth values among nodes.	Two-step scheduling: static, then including public cloud to address deadline.	Initial scheduling step considers private resources; public resources are used if necessary.	DAG with compute cost based on number of instructions.
VECCHIOLA et al., 2012	Not specified.	Not addressed.	Public resources used if necessary.	Supported, but no details provided.

In terms of dynamic scheduling the authors claim that when it is not possible to assign tasks to resources due to resource unavailability, the recomputation phase of PSO dynamically balances other tasks' mappings. However, there is no explicit mention to dynamically (re)scheduling based on other aspects, such as performance fluctuations and reliability issues. Workflow support is limited to the usual DAG-based description wherein computation costs of a task on a compute host is a known information and edges represent the communication among phases. This representation provides a limited amount of information regarding the workflow, such as performance fluctuation due to branches and other logic, requirements related to memory and local storage, and the actual performance observed when executing one of the phases on a node.

Lin and Lu [64] propose an algorithm named SHEFT, Scalable HEFT (Heterogeneous Earliest Finish Time). The authors claim that resources within one cluster usually share the same network communication, so they have the same data transfer rate with each other. While there might be network utilization fluctuations during the execution of a workflow (and even in idle state) that invalidate this assumption, the fact is that even locally (in the same node) there is data access imbalance due to contention – concurrency to access the same resources, in this case I/O. For example, if two containers (or virtual machines) located in the same node attempt to access a file or a network stream, they will naturally compete for resources. There is not clear support to dynamic scheduling to address reliability-related issues or performance fluctuations. The solution supports workflows but there are no details on how workflows are modeled or mapped into execution space.

Xu et al. [133] propose MQMW, a Multiple QoS constrained scheduling strategy of Multi-Workflows. Four factors that affect makespan and cost are selected: available service number, time and cost covariance, time quota, and cost quota. Workflows are modeled as DAGs but no specific information about the modeling is provided. The approach adopted by the authors to support multiple workflows is based on the creation of composite DAGs representing multiple workflows. DAG nodes with no predecessors (e.g., input nodes) are connected to a common entry node shared by multiple workflows. In this sense, new workflows to be executed are joined via a single merging point. Finally, there is no explicit support to dynamic scheduling or heterogeneous environments.

Weissman and Grimshaw [126] propose a scheduling solution for heterogeneous environments (wide-area systems) that encompasses data-intensive and dynamic scheduling properties. The solution also maintains local autonomy for scheduling decisions – remote resources are explored only when appropriate. Moreover, according to the authors the unpredictability of resource sharing in large distributed areas requires scheduling to be deferred until runtime. For data-intensive properties, it is assumed that the system infrastructure is able to access data and files independent of location. If data needs to be transported (e.g., jobs scheduled in a site that does not have direct access to needed data), the scheduling system assumes that data transport cost can be amortized over the course of job execution. This is not always possible as even local transfers can be expensive, especially if multiple local workers shared the same resources – a common scenario for cloud environments, with a high density of worker elements per physical node.

Chen and Zhang [23] use the Ant Colony Optimization (ACO) metaheuristic that simulates the pheromone depositing and following behavior of ants and it is applied to numerous intractable combinatorial optimization problems. QoS parameters are based on reliability, makespan, and cost. Reliability is defined as the minimum reliability of all selected service instances in the workflow. The actual reliability aspects or metrics used in the calculations, however, are not disclosed. Data communication and transfers are not explicitly addressed in the paper.

Rodriguez and Buyya [95] propose a resource provisioning and scheduling solution for execution of scientific workflows on cloud infrastructures. The solution is based on particle swarm optimization aiming at minimizing execution cost while meeting deadline constraints. The general approach adopted by the authors is similar to the one from [82]. Virtual machines are assumed to have a fixed compute capacity (measured in FLOPS), although some degree of performance variation due to degradation is considered in their model. In addition, the authors assume that workflows are executed on a single data center or region, and as a consequence the bandwidth between each virtual machine should be roughly the same. However, this might not be true even for a set of nodes connected to the same switch, especially during phases wherein several demanding data transfers are executed among nodes – for example, when inputs are distributed to all worker nodes. Finally, the transfer cost between two tasks being executed on the same virtual machine is assumed to be zero, while the actual communication can be much more expensive than that, especially if it is via file I/O. The workflow modeling is based on a DAG with fixed transfer costs (edges). Task costs are calculated based on the size of the task measured in FLOPS. The cost of a task, consequently, depends on the computational complexity of this task instead of the input data. Of course the number of FLOPS can be calculated based on the size of the input data, but no remarks are made in that sense. No other properties are defined, such as performance variation due to branching and input sizes.

Fard et al. [33] propose a multi-objective scheduling solution and present a case study comprising makespan, cost, energy, and reliability. The workflow is modeled as a very simple DAG with fixed size data dependencies among tasks. Nodes are modeled as a mesh network wherein each point-to-point connection has a different bandwidth. Cost is modeled as a sum of computation, storage, and transfer costs. Energy consumption is modeled only after the compute phases of the workflow. The authors state that their focus is on computational-intensive applications, thus only the computation part of the activities are considered in the energy consumption calculation, while "data transfers and storage time are ignored". Finally, reliability is modeled using an exponential distribution representing the probability of successful completion of a task.

Malawski et al. [70] investigate the management of workflow ensembles under budget and deadline constraints on clouds. The authors state that although workflows are often data-intensive, the algorithms described do not consider the size of input and output data when scheduling tasks". In other words, the scheduling cost is uniquely based on computation time. The authors complement by stating that data is stored in a shared cloud storage system and that intermediate data transfer times are included in task run times – transfer time is modeled as part of computation time. It is also assumed that data transfer times between the shared storage and the VMs are equal for different VMs so that task placement decisions do not impact the runtime of the tasks. It is clear, then, that any issues related to contention, performance variation due to network and I/O bandwidth utilization shared among several worker nodes and virtual machines, and the impact of sequentially distributing input among workers are partially or entirely overlooked depending on the case.

Sakellariou and Zhao [97] propose a scheduling mechanisms that considers executing carefully selected rescheduling operations to achieve better performance without imposing a large overhead compared to solutions that dynamically attempt to reschedule before the execution of every task. While the proposal is designed for grid computing, the ideas related to the selection of points of interest to execute the rescheduling operation is relevant also for cloud environments. The resource and workflow models adopted by the authors imply a fundamental simplification of how computation and transfer costs are calculated. Each task has a different cost for each machine, expressed as time per data unit. Although this attempts to model performance differences between nodes, this implies that the computation cost of each task linearly varies with the amount of input data. In contrast, if the assumption is that the costs are expressed as a fixed amount, then they are simply fixed to a value assuming a certain amount of input. Both cases do not consider a more sophisticated workflow model in which computation and communication costs vary according to the size of input data not linearly, but expressed as a general function that can be either predefined or dynamically obtained. This modeling affects both the initial static schedule and also subsequent rescheduling operations.

Wang and Chen [124] propose a cost function that considers the robustness of a schedule regarding the probability of successful execution. Based on the paper, failure is considered to be any event that leads to abnormal termination of a task, and consequent loss of all workflow progress thus far. Afterwards the cost function is used in conjunction with a genetic algorithm to find an optimized schedule that maximizes its robustness. However, in the definition of the cost of failure function the authors assume that the potential loss in the execution cost of each task is independent of the other workflow tasks. In other words, a failure always has a local scope, without possibility of chaining impact outside the workflow. Moreover, there is no workflow characterization in terms of data transfers and task costs. Robustness or failure rates are not specified or tied to a specific property such as MTBF (Mean Time Between Failures).

Poola et al. [86] propose a fault-tolerant workflow scheduling using spot and on-demand cloud instances to reduce execution cost and meet workflow deadlines. Workflow model is based on a DAG. Data transfer times are accounted for with a model based on the data size and the cloud data center internal bandwidth (assumed to be fixed for all nodes). Task execution time is estimated based on the information of number of instructions of the task. For fault-tolerance the authors adopt checkpointing, which consists of creating snapshots of the data being manipulated by the workflow and run time structures, if necessary. The core idea is to store enough information to restart computation in case of an error. One of the issues with the approach adopted is how checkpointing is considered in the model. Checkpointing worst-case scenario requires a full memory dump, meaning that 100% of the memory contents have to be written to a persistent storage (e.g., spinning disks). Depending on the memory footprint of the workflow phase this amount surpass the order of gigabytes. However, in the model proposed in the paper the checkpointing cost is not considered "as the price of storage service is negligible compared to the cost of VMs". Moreover, while checkpointing time was considered in their model, the actual checkpointing time on spinning disks, especially for cloud systems that are not specialized for parallel I/O, can represent much more than 10% of overhead, which is the value expected for very large-scale machines such as APEX and EXASCALE. Thus, either the checkpointing size adopted is much smaller than what is observed for real scientific workflow or the checkpointing mechanism is creating partial checkpoints. Nevertheless, the results obtained by the authors show that having checkpoints actually reduces the final cost. Yet, the fault-tolerance provided by the method only covers the repair part, not the fault avoidance part. There is no (explicit) logic to predict the probability of occurrence of failures due to some hardware or software property, for instance.

Bittencourt and Madeira [15] propose HCOC, the Hybrid Cloud Optimized Cost, a scheduling algorithm that selects the resources to be leased from a public cloud to complement the resources from a private cloud. The objective of HCOC is to reduce makespan to fit a desired execution time or deadline while maintaining a reasonable cost. This cost constraint is introduced to limit the amount of resources leased from the public cloud, otherwise the public cloud would always be overutilized to address the time constraints. Intra-node communication is considered to be limitless, in the sense that the costs of local communication are ignored. Communication cost is calculated by dividing the amount of data by the link bandwidth, which is modeled as a constant value. Computation cost is based on the number of instructions and the processing capacity of a node, which is measured as instructions per time. There are several implicit assumptions in this model, such as fixed capacity for transferring and computing. There is not a function that varies the amount of computation based on the size of the input.

Vecchiola et al. [120] claim that scientific applications require a large computing power that typically exceeds the resources of a single institution. In this sense, their solution aims at providing a deadline-based provisioning mechanisms for hybrid clouds, allowing the combination of local resources to the ones obtained from a public cloud service. However, there are no specific details on how workflows are internally handled by their solution, nor how resources are mapped to workflow phases or how costs are calculated. Moreover, their solution (named Aneka) focuses on meeting a specific deadline, thus not addressing issues related to total execution time (makespan) or reliability.

Gaps and challenges

This section discusses the gaps and challenges identified in the investigation of related work.

Data-intensive loads

Regarding data-intensive loads, [82] states that they represent a special class of applications where the size and/or quantity of data is large. As a direct result, transfer costs are significantly higher and more prominent. While the authors do address data transfers in their resource model, several aspects of data access are not acknowledged. For instance, accesses to the same resource leads to a communication cost of zero. Transfer costs are calculated based on average bandwidth between the nodes, without regards to I/O contention, multiples accesses to the same resource, containers and VMs co-located in the same node sharing network and I/O resources, among other factors. This is also observed in other works such as [15, 33, 64, 95]. Other models consider transfers as part of computation time, such as [70]. This is depicted as a fundamental challenge by [127], which states that "in most studies, data transfer between tasks is not directly considered, data uploading and downloading are assumed as part of task execution". Wu et al. [127] complements by stating that this may not be the case for modern applications and workflows –in fact, data movement activities might dominate both execution time and cost. For the authors it is essential to design the data placement strategies for resource provisioning decision-making. Moreover, employing VMs deployed in different regions intensifies the data transfer costs, leading to an even more complicated issue. This is correlated to having more complex cloud environments in terms of resource distribution, such as hybrid and multicloud scenarios.

Hybrid and multicloud scenarios

Regarding hybrid and multicloud scenarios, [127] states that it is necessary hybrid environments, heterogeneous resources, and multicloud environments. Singh and Chana [109] also highlights the importance of hybrid and multicloud scenarios for future deployments of large-scale cloud environments and reach performance comparable to large-scale scientific clusters. On the other hand, most of the scheduling solution still do not address hybrid clouds nor multiclouds. The few ones that do implement mechanisms that use the public part of a hybrid cloud to lease additional resources if necessary – the hybrid component of the setup is treated as a supporting element, not as protagonist. For example, [15] and [120] propose solutions that only allocate resources from the hybrid cloud (the public part of it) if the private part is not able to handle the workflow execution. Multicloud support is even more scarce or not explicit. Several of the proposed solutions could be adopted or adapted to multicloud environments, but there still is a lack of experimental results to match the predicted importance of such large-scale setups.

The motivation for multicloud environments vary from having more raw performance to match other large-scale deployments to having more options in terms of available services. Simarro et al. [107], for instance, states that resource placements across several cloud offers are useful to obtain resources at the best cost ratio. The same approach is adopted by [37] and [101]. Regarding the execution of large-scale applications on similar scale systems, [68] suggest a multi-site workflow scheduling technique to enhance the range of available resources to execute workflows. While their approach does consider data transfers and the costs of sending data over expensive (slower) links that connect different geographically distributed sites, their approach does not consider 1) performance fluctuations during execution of the workflow, which would suggest the implementation of rescheduling and rebalancing mechanisms; 2) reliability mechanisms to cope with performance fluctuations due to failures; and 3) the influence of contention in the general I/O operations, such as sequential accesses to the same data inputs.

Rescheduling and performance fluctuations

Performance fluctuations caused by multi-tenant resource sharing is one of the major components that must be included in the definition of uncertainties associated to scheduling operations [127]. The authors complement: "The most important problem when implementing algorithms in real environment is the uncertainties of task execution and data transmission time". Moreover, most works assume that a workflow has a definite DAG structure while actual workflows have loops and conditional branches. For instance, the execution control in several scientific workflows is based on conditions that are calculated every iteration, meaning that branches are essential to determine whether the pipelines must be stopped or not. In this sense, rescheduling techniques are usually adopted to correct potential deviations from an original guess of the performance of a workflow on a system [61, 127].

Reliability

Several authors and works highlight the challenges and potential gaps in terms of cloud management and cloud resource management in terms of reliability. Bala and Chana [9] states that workflow scheduling is one of the key issues in the management of workflow execution in cloud environments and that existing scheduling algorithms (at least at that time) did not consider reliability and availability aspects in the cloud environment. Singh and Chana [109] directly addressed this issue by stating that the hardware layer must be reliable before allocating resources. While several subsequent works addressed these aspects, there still are gaps in the methodology. For instance, [23] implement a solution that considers a reliability factor but there is no explicit model on how to calculate this factor based on actual hardware and software reliability related metrics, such as hardware failure and software interruption rates.

Fard et al. [33] defines a reliability factor by assuming a statistically independent constant failure rate, but this rate only reflects the probability of successful completion of a task – there is no clear connection between this concept and a factual and measurable metric from hardware and software point of view. Hakem and Butelle [43] also proposes a reliability-based resource allocation solution by defining a reliability model divided in processor, link, and system. The model is based on exponential distributions which could be related to metrics such as mean time between failures (MTBF) and failure in time (FIT).

Other solutions such as the one from [87] use reliability-related methods such as checkpointing to decrease application failures, but in this particular case, for instance, the performance implications of having these mechanisms is not fully appreciated. The I/O cost in terms of storage and time to implement checkpointing are far from negligible. Still on reliability, [124] state that the main two strategies to calculate reliability factors is to either establish a reputation threshold or to treat nodes independently and multiply their probability of success. Still, the reliability approach proposed by the authors does not address measurable metrics to calculate these factors. Moreover, on one side there are the solutions only address failures after their occurrence, not before. For instance, [86] uses checkpointing to recover from failures but there is no mechanism in place to calculate the probability of failures and attempt to avoid nodes with higher probability of failure, or at least designate a smaller portion of tasks to this node. On the other side, solutions calculate reliability factors based on theoretical metrics that might not reflect the specificities of each node and there are no clear mechanism to combine prevention and recovery. In that sense, [49] provides a deeper analysis of fault-tolerance techniques for grid computing that could be applied to cloud computing. The authors clearly state that the requirements for implementing failure recovery mechanisms on grids comprise support for diverse failure handling strategies, separation of failure handling policies from application codes, and user-defined exception handling. In terms of task-level failure handling techniques the authors consider retrying (straightforward and potentially least efficient of the enlisted techniques), replication (replicas running on different resources), and checkpointing. Checkpointing is actively used in real scientific scenarios while replication usually leads to prohibitive costs, as in several cases running one replica is expensive enough in terms of resource demand. In addition, in terms of workflow-level failure handling, the authors propose mechanisms such as alternative task (try a different implementation when available), workflow-level redundancy, and user-defined exceptions that are able to fallback to reliable failure handling. In terms of evaluation the authors propose parameters such as failure-free execution time, failure rates, downtime, recovery time, checkpointing overhead, among others. These are measurable metrics that can be used to model and represent the failure behavior of systems and workflows.

Conclusion

This paper provided an extensive investigation of existing works in cloud resource management. The investigation started by providing several definitions and associated concepts on the subject, covering the rationale presented by several authors and publications from the academia. Three main works were selected in this sense, reflecting the works that provided a clear definition of distinct steps regarding cloud resource management. Among these works the common point is the association of management components to each phase of the resource lifecycle, such as resource discovery, allocation, scheduling, and monitoring. Moreover, the ultimate objective in all cases is to enable task execution while optimizing infrastructural efficiency. These are the two main points related to cloud resource management.

The next step in this investigation was to identify relevant works in the area, focusing on recent publications and others not so recent but still important, for instance covering a specific aspect of cloud resource management. The results of this analysis led to the identification of over 110 works on cloud resource management. A taxonomy was created based on the consolidation of characteristics and properties used to classify the selected works. Further analysis was provided to enhance the identification of gaps and challenges for future research on cloud resource management focusing on large-scale applications and workflows. The final step of this investigation was the formalization of these gaps and challenges obtained during the research. The challenges were organized in four topics: a) challenges related to data-intensive workflows, including lack of proper modeling of transfers, or modeling of transfers as part of computation; b) hybrid and multicloud scenarios, comprising large-scale deployments and more complex setups in terms of resource distribution; c) rescheduling and performance fluctuations, essentially addressing the lack of mechanisms to adequately cope with the inherent performance fluctuation of large scale cloud deployments, and the effects of multi-tenancy and resource sharing; and d) reliability, highlighting the lack of proper factors based on actual and measurable metrics such as failure rates. Based on these topics, four clear gaps are identified to be addressed by future research:

- Lack of mechanisms to address the particularities of data-intensive workflows, especially considering that future trends point to the direction of I/O workflows with intensive data movement and with reliability-related mechanisms highly dependent on I/O as well.
- Lack of mechanisms to address the particularities of large-scale cloud setups with more complex environments in terms of resource heterogeneity and distribution, such as hybrid and multicloud scenarios, which are expected to be the main drivers for large-scale utilization of cloud – scientific workflows being one important instance.
- Lack of mechanisms to address the fluctuations in workflow progress due to performance variation and reliability, both phenomena that can be partially or even fully addressed by implementing controlled rescheduling policies.
- Lack of reliability mechanisms based on actual and measurable metrics that can be derived from documentation and from collecting information of the system.

The results of this analysis combined to the requirements identified for future workloads leads to the conclusion that modern solutions aiming at providing resource management for large-scale deployments and to execute large-scale problems must provide mechanisms to address data movement in massive scale while adequately distributing resources to tasks, adjusting this distribution depending on the fluctuations observed in the system. Existing solutions can and should be adapted to address the specific requirements related to the challenges identified, but further research and development are necessary to cope with these requirements in a more comprehensive and decisive way.

Authors' contributions

NMG carried out the survey of the literature, created the taxonomy, analyzed the references, drafted the manuscript, and identified th open issues and future challenges. TCMBC and CCM provided insights and guidance in developing the taxonomy as well as in the analysis of the references and identification of gaps. All authors read and approved the manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

[1]Escola Politécnica, University of São Paulo (USP), São Paulo/SP, Brazil. [2]Santa Catarina State University (UDESC), Joinville/SC, Brazil.

Received: 17 November 2016 Accepted: 18 May 2017
Published online: 19 June 2017

References

1. Abrishami S, Naghibzadeh M, Epema DH (2013) Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds. Futur Gener Comput Syst 29(1):158–169
2. Anand L, Ghose D, Mani V (1999) Elisa: an estimated load information scheduling algorithm for distributed computing systems. Comput Math Appl 37(8):57–85
3. Andrade N, Cirne W, Brasileiro F, Roisenberg P (2003) Ourgrid: An approach to easily assemble grids with equitable resource sharing. In: Workshop on Job Scheduling Strategies for Parallel Processing. Springer, Berlin. pp 61–86
4. Arabnejad H, Barbosa JG (2014a) A budget constrained scheduling algorithm for workflow applications. J Grid Comput 12(4):665–679
5. Arabnejad H, Barbosa JG (2014b) List scheduling algorithm for heterogeneous systems by an optimistic cost table. IEEE Trans Parallel Distrib Syst 25(3):682–694
6. Arabnejad V, Bubendorfer K Cost effective and deadline constrained scientific workflow scheduling for commercial clouds. In: Network Computing and Applications (NCA), 2015 IEEE 14th International Symposium On. pp 106–113. doi:10.1109/NCA.2015.33
7. Armbrust M, Fox A, Griffith R, Joseph AD, Katz RH, Konwinski A, Lee G, Patterson DA, Rabkin A, Stoica I, Zaharia M (2010) A View of Cloud Computing. Commun. ACM, New York. 53(4):50–58. Technical Report No. UCB/EECS-2009-28. Available on: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html, http://doi.acm.org/10.1145/1721654.1721672, doi:10.1145/1721654.1721672

8. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, et al (2010) A view of cloud computing. Commun ACM 53(4):50–58
9. Bala A, Chana I (2011) Article: A Survey of Various Workflow Scheduling Algorithms in Cloud Environment. In: IJCA Proceedings on 2nd National Conference on Information and Communication Technology, Nagpur Vol. NCICT. pp 26–30
10. Bellavista P, Corradi A, Kotoulas S, Reale A (2014) Adaptive fault-tolerance for dynamic resource provisioning in distributed stream processing systems. In: Proceedings of 17th International Conference on Extending Database Technology (EDBT), March 24-28, 2014, Athens, Greece: ISBN 978-3-89318065-3, on OpenProceedings.org. Open Proceedings.org, Athens. pp 85–96. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.673.2146
11. Berl A, Gelenbe E, Di Girolamo M, Giuliani G, De Meer H, Dang MQ, Pentikousis K (2010) Energy-efficient cloud computing. Comput J 53(7):1045–1051
12. Bessai K, Youcef S, Oulamara A, Godart C, Nurcan S (2012) Bi-criteria workflow tasks allocation and scheduling in cloud computing environments. In: Cloud Computing (CLOUD), 2012 IEEE 5th International Conference On. IEEE, Honolulu. pp 638–645
13. Bharathi S, Chervenak A (2009) Data staging strategies and their impact on the execution of scientific workflows. In: Proceeding DADC '09 Proceedings of the Second International Workshop on Data-aware Distributed Computing. ACM, New York. p 5
14. Bilgaiyan S, Sagnika S, Das M (2014) Workflow scheduling in cloud computing environment using cat swarm optimization. In: Advance Computing Conference (IACC), 2014 IEEE International. IEEE, Gurgaon. pp 680–685
15. Bittencourt LF, Madeira ERM (2011) Hcoc: a cost optimization algorithm for workflow scheduling in hybrid clouds. J Internet Serv Appl 2(3):207–227
16. Butt AR, Zhang R, Hu YC (2003) A self-organizing flock of condors. In: Proceedings of the 2003 ACM/IEEE Conference on Supercomputing. ACM, Phoenix. p 42
17. Buyya R, Yeo CS, Venugopal S (2008) Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In: High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference On. IEEE, Dalian. pp 5–13. doi:10.1109/HPCC.2008.172
18. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I (2009) Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. Futur Gener Comput Syst 25(6):599–616
19. Byun EK, Kee YS, Kim JS, Maeng S (2011) Cost optimized provisioning of elastic resources for application workflows. Futur Gener Comput Syst 27(8):1011–1026
20. Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, Buyya R (2011) Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw Pract Experience 41(1):23–50
21. Chana I, Singh S (2014) Quality of Service and Service Level Agreements for Cloud Environments: Issues and Challenges. In: Mahmood Z (ed). Cloud Computing: Challenges, Limitations and R&D Solutions. Springer International Publishing, Switzerland. pp 51–72. doi:10.1007/978-3-319-10530-7_3. http://dx.doi.org/10.1007/978-3-319-10530-7_3, https://link.springer.com/chapter/10.1007%2F978-3-319-10530-7_3
22. Chard R, Chard K, Bubendorfer K, Lacinski L, Madduri R, Foster I (2015) Cost-aware elastic cloud provisioning for scientific workloads. In: 2015 IEEE 8th International Conference on Cloud Computing. pp 971–974. doi:10.1109/CLOUD.2015.130
23. Chen WN, Zhang J (2009) An ant colony optimization approach to a grid workflow scheduling problem with various qos requirements. IEEE Trans Syst Man Cybern C (Appl Rev) 39(1):29–43
24. Chieu TC, Mohindra A, Karve AA, Segal A (2009) Dynamic scaling of web applications in a virtualized cloud computing environment. In: e-Business Engineering, 2009. ICEBE'09. IEEE International Conference On. IEEE, Washington, DC. pp 281–286
25. Cordasco G, Malewicz G, Rosenberg AL (2010) Extending ic-scheduling via the sweep algorithm. J Parallel Distrib Comput 70(3):201–211
26. Dastjerdi AV, Buyya R (2012) An autonomous reliability-aware negotiation strategy for cloud computing environments. In: Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium On. IEEE, Ottawa. pp 284–291
27. Daval-Frerot C, Lacroix M, Guyennet H (2000) Federation of resource traders in objects-oriented distributed systems. In: Proceedings of the International Conference on Parallel Computing in Electrical Engineering. IEEE Computer Society, Washington, DC. p 84
28. Demchenko Y, Blanchet C, Loomis C, Branchat R, Slawik M, Zilci I,Bedri M, Gibrat JF, Lodygensky O, Zivkovic M, d. Laat C (2016) Cyclone: A platform for data intensive scientific applications in heterogeneous multi-cloud/multi-provider environment. In: 2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW). pp 154–159. doi:10.1109/IC2EW.2016.46
29. Dias de Assunção M, Buyya R, Venugopal S (2008) Intergrid: A case for internetworking islands of grids. Concurr Computat Pract Experience 20(8):997–1024
30. Dogan A, Ozguner F (2002) Matching and scheduling algorithms for minimizing execution time and failure probability of applications in heterogeneous computing. IEEE Trans Parallel Distrib Syst 13(3):308–323
31. Dongarra JJ, Jeannot E, Saule E, Shi Z (2007) Bi-objective scheduling algorithms for optimizing makespan and reliability on heterogeneous systems. In: Proceedings of the Nineteenth Annual ACM Symposium on Parallel Algorithms and Architectures. ACM, San Diego. pp 280–288
32. Fard HM, Prodan R, Fahringer T (2013) A truthful dynamic workflow scheduling mechanism for commercial multicloud environments. IEEE Trans Parallel Distrib Syst 24(6):1203–1212
33. Fard HM, Prodan R, Barrionuevo JJD, Fahringer T (2012) A multi-objective approach for workflow scheduling in heterogeneous environments. In: Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012). IEEE Computer Society, Ottawa. pp 300–309
34. Felter W, Ferreira A, Rajamony R, Rubio J (2015) An updated performance comparison of virtual machines and linux containers. In: Performance Analysis of Systems and Software (ISPASS), 2015 IEEE International Symposium On. IEEE, Philadelphia. pp 171–172
35. Fölling A, Grimme C, Lepping J, Papaspyrou A (2009) Decentralized grid scheduling with evolutionary fuzzy systems. In: Workshop on Job Scheduling Strategies for Parallel Processing. Springer, Rome. pp 16–36
36. Foster I, Zhao Y, Raicu I, Lu S (2008) Cloud computing and grid computing 360-degree compared. In: 2008 Grid Computing Environments Workshop. IEEE, Austin. pp 1–10
37. Frincu ME, Craciun C (2011) Multi-objective meta-heuristics for scheduling applications with high availability requirements and cost constraints in multi-cloud environments. In: Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference On. IEEE, Victoria. pp 267–274
38. Gao Y, Wang Y, Gupta SK, Pedram M (2013) An energy and deadline aware resource provisioning, scheduling and optimization framework for cloud systems. In: Proceedings of the Ninth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis. IEEE Press, Montreal. p 31
39. Garg SK, Buyya R, Siegel HJ (2010) Time and cost trade-off management for scheduling parallel applications on utility grids. Futur Gener Comput Syst 26(8):1344–1355
40. Garg SK, Gopalaiyengar SK, Buyya R (2011) Sla-based resource provisioning for heterogeneous workloads in a virtualized cloud datacenter. In: International Conference on Algorithms and Architectures for Parallel Processing. Springer, Melbourne. pp 371–384
41. Grekioti A, Shakhlevich NV (2013) Scheduling bag-of-tasks applications to optimize computation time and cost. In: International Conference on Parallel Processing and Applied Mathematics. Springer, Warsaw. pp 3–12
42. Grewal RK, Pateriya PK (2013) A Rule-Based Approach for Effective Resource Provisioning in Hybrid Cloud Environment(Patnaik S,Tripathy P, Naik S, eds.). Springer, Berlin, Heidelberg, pp 41–57
43. Hakem M, Butelle F (2007) Reliability and scheduling on systems subject to failures. In: 2007 International Conference on Parallel Processing (ICPP 2007). IEEE, XiAn. pp 38–38
44. He S, Guo L, Guo Y, Wu C, Ghanem M, Han R (2012) Elastic application container: A lightweight approach for cloud resource provisioning. In: 2012 IEEE 26th International Conference on Advanced Information Networking and Applications. IEEE, Fukuoka. pp 15–22

45. Hofmann P, Woods D (2010) Cloud computing: The limits of public clouds for business applications. IEEE Internet Comput 14(6):90–93. doi:10.1109/MIC.2010.136
46. Huang Y, Bessis N, Norrington P, Kuonen P, Hirsbrunner B (2013) Exploring decentralized dynamic scheduling for grids and clouds using the community-aware scheduling algorithm. Futur Gener Comput Syst 29(1):402–415
47. Hwang E, Kim KH (2012a) Minimizing cost of virtual machines for deadline-constrained mapreduce applications in the cloud. In: 2012 ACM/IEEE 13th International Conference on Grid Computing. IEEE, Beijing. pp 130–138
48. Hwang E, Kim KH (2012b) Minimizing cost of virtual machines for deadline-constrained mapreduce applications in the cloud. In: 2012 ACM/IEEE 13th International Conference on Grid Computing. IEEE, Beijing. pp 130–138
49. Hwang S, Kesselman C (2003) Grid workflow: a flexible failure handling framework for the grid. In: High Performance Distributed Computing, 2003. Proceedings. 12th IEEE International Symposium On. IEEE, Seattle. pp 126–137
50. Iqbal W, Dailey MN, Carrera D, Janecek P (2011) Adaptive resource provisioning for read intensive multi-tier applications in the cloud. Futur Gener Comput Syst 27(6):871–879
51. Iverson MA, et al. (1999) Hierarchical, competitive scheduling of multiple dags in a dynamic heterogeneous environment. J Distrib Syst Eng 9 The British Computer Society. United Kingdom, IOP, Bristol 6(3):112–120
52. Jennings B, Stadler R (2015) Resource management in clouds: Survey and research challenges. J Netw Syst Manag 23(3):567–619
53. Kacamarga MF, Pardamean B, Wijaya H (2015) Lightweight virtualization in cloud computing for research. In: International Conference on Soft Computing, Intelligence Systems, and Information Technology. Springer, Bali. pp 439–445
54. Kertesz A, Kecskemeti G, Brandic I (2011) Autonomic sla-aware service virtualization for distributed systems. In: 2011 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing. IEEE, Ayia Napa. pp 503–510
55. Kim KH, Buyya R, Kim J (2007) Power aware scheduling of bag-of-tasks applications with deadline constraints on dvs-enabled clusters. In: CCGRID Vol. 7. pp 541–548
56. Kousiouris G, Menychtas A, Kyriazis D, Gogouvitis S, Varvarigou T (2014) Dynamic, behavioral-based estimation of resource provisioning based on high-level application terms in cloud platforms. Futur Gener Comput Syst 32:27–40
57. Kwok YK, Ahmad I (1996) Dynamic critical-path scheduling: An effective technique for allocating task graphs to multiprocessors. IEEE Trans Parallel Distrib Syst 7(5):506–521
58. Lai K, Rasmusson L, Adar E, Zhang L, Huberman BA (2005) Tycoon: An implementation of a distributed, market-based resource allocation system. Multiagent Grid Syst 1(3):169–182
59. Leal K, Huedo E, Llorente IM (2009) A decentralized model for scheduling independent tasks in federated grids. Futur Gener Comput Syst 25(8):840–852
60. Lee YC, Zomaya AY (2011) Energy conscious scheduling for distributed computing systems under different operating conditions. IEEE Trans Parallel Distrib Syst 22(8):1374–1381
61. Lee YC, Subrata R, Zomaya AY (2009) On the performance of a dual-objective optimization model for workflow applications on grid platforms. IEEE Trans Parallel Distrib Syst 20(9):1273–1284
62. Li J, Su S, Cheng X, Huang Q, Zhang Z (2011) Cost-conscious scheduling for large graph processing in the cloud. In: High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference On. pp 808–813. doi:10.1109/HPCC.2011.147
63. Li XY, Zhou LT, Shi Y, Guo Y (2010) A trusted computing environment model in cloud architecture. In: 2010 International Conference on Machine Learning and Cybernetics. IEEE, Qingdao Vol. 6. pp 2843–2848
64. Lin C, Lu S (2011) Scheduling scientific workflows elastically for cloud computing. In: Cloud Computing (CLOUD), 2011 IEEE International Conference On. IEEE, Washington. pp 746–747
65. Lin X, Wu CQ (2013) On scientific workflow scheduling in clouds under budget constraint. In: 2013 42nd International Conference on Parallel Processing. IEEE, Lyon. pp 90–99
66. Liu D, Zhao L (2014) The research and implementation of cloud computing platform based on docker. In: Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2014 11th International Computer Conference On. IEEE, Chengdu. pp 475–478
67. Liu K, Jin H, Chen J, Liu X, Yuan D, Yang Y (2010) A compromised-time-cost scheduling algorithm in swindew-c for instance-intensive cost-constrained workflows on cloud computing platform. Int J High Perform Comput Appl
68. Maheshwari K, Jung ES, Meng J, Morozov V, Vishwanath V, Kettimuthu R (2016) Workflow performance improvement using model-based scheduling over multiple clusters and clouds. Futur Gener Comput Syst 54:206–218
69. Majumdar S (2011) Resource management on clouds and grids: challenges and answers. In: Proceedings of the 14th Communications and Networking Symposium. Society for Computer Simulation International, Boston. pp 151–152
70. Malawski M, Juve G, Deelman E, Nabrzyski J (2012) Cost-and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds. In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. IEEE Computer Society Press, Salt Lake City. p 22
71. Malewicz G, Foster I, Rosenberg AL, Wilde M (2007) A tool for prioritizing dagman jobs and its evaluation. J Grid Comput 5(2):197–212
72. Manvi SS, Shyam GK (2014) Resource management for infrastructure as a service (iaas) in cloud computing: A survey. J Netw Comput Appl 41:424–440
73. Mao M, Humphrey M (2011) Auto-scaling to minimize cost and meet application deadlines in cloud workflows. In: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis. ACM, Seattle. p 49
74. Mao M, Li J, Humphrey M (2010) Cloud auto-scaling with deadline and budget constraints. In: 2010 11th IEEE/ACM International Conference on Grid Computing. IEEE, Brussels. pp 41–48
75. Marinescu DC (2013) Cloud Computing: Theory and Practice. Morgan Kauffman, Waltham
76. Mell P, Grance T (2011) The nist definition of cloud computing. Gaithersburg
77. Merkel D (2014) Docker: lightweight linux containers for consistent development and deployment. Linux J 2014(239):2
78. Mezmaz M, Melab N, Kessaci Y, Lee YC, Talbi EG, Zomaya AY, Tuyttens D (2011) A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems. J Parallel Distrib Comput 71(11):1497–1508
79. Mishra R, Rastogi N, Zhu D, Mossé D, Melhem R (2003) Energy aware scheduling for distributed real-time systems. In: Parallel and Distributed Processing Symposium, 2003. Proceedings. International. IEEE, Nice. p 9
80. Mustafa S, Nazir B, Hayat A, Madani SA, et al. (2015) Resource management in cloud computing: Taxonomy, prospects, and challenges. Comput Electr Eng 47:186–203
81. Nurmi D, Wolski R, Grzegorczyk C, Obertelli G, Soman S, Youseff L, Zagorodnov D (2009) The eucalyptus open-source cloud-computing system. In: Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium On. IEEE, Shanghai. pp 124–131
82. Pandey S, Wu L, Guru SM, Buyya R (2010) A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In: 2010 24th IEEE International Conference on Advanced Information Networking and Applications. IEEE, Perth. pp 400–407
83. Park SM, Humphrey M (2008) Data throttling for data-intensive workflows. In: 2008 IEEE International Symposium on Parallel and Distributed Processing. IEEE, Miami. pp 1–11. http://ieeexplore.ieee.org/abstract/document/4536306/, doi:10.1109/IPDPS.2008.4536306
84. Parsa S, Entezari-Maleki R (2009) Rasa: A new task scheduling algorithm in grid environment. World Appl Sci J 7:152–160
85. Phaphoom N, Wang X, Abrahamsson P (2013) Foundations and technological landscape of cloud computing. ISRN Softw Eng
86. Poola D, Ramamohanarao K, Buyya R (2014a) Fault-tolerant workflow scheduling using spot instances on clouds. Procedia Comput Sci 29:523–533
87. Poola D, Garg SK, Buyya R, Yang Y, Ramamohanarao K (2014b) Robust scheduling of scientific workflows with deadline and budget constraints

in clouds. In: 2014 IEEE 28th International Conference on Advanced Information Networking and Applications. IEEE, Victoria. pp 858–865

88. Prodan R, Wieczorek M (2010) Bi-criteria scheduling of scientific grid workflows. IEEE Transactions on Automation Science and Engineering 7(2):364–376
89. Pruhs K, van Stee R, Uthaisombut P (2008) Speed scaling of tasks with precedence constraints. Theory of Computing Systems 43(1):67–80
90. Ramakrishnan A, Singh G, Zhao H, Deelman E, Sakellariou R, Vahi K, Blackburn K, Meyers D, Samidi M (2007) Scheduling data-intensiveworkflows onto storage-constrained distributed resources. In: Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid'07). IEEE, Rio de Janeiro. pp 401–409
91. Ren K, Wang C, Wang Q (2012) Security challenges for the public cloud. IEEE Internet Comput 16(1):69
92. Rittinghouse J, Ransome J (2009) Cloud Computing: Implementation, Management, and Security. 1st edn. CRC Press, Inc., Boca Raton, FL, USA
93. Rodero I, Guim F, Corbalan J (2009) Evaluation of coordinated grid scheduling strategies. In: High Performance Computing and Communications, 2009. HPCC'09. 11th IEEE International Conference On. IEEE, Seoul. pp 1–10
94. Rodero I, Guim F, Corbalan J, Fong L, Sadjadi SM (2010) Grid broker selection strategies using aggregated resource information. Futur Gener Comput Syst 26(1):72–86
95. Rodriguez MA, Buyya R (2014) Deadline based resource provisioningand scheduling algorithm for scientific workflows on clouds. IEEE Trans Cloud Comput 2(2):222–235
96. Rosenberg F, Celikovic P, Michlmayr A, Leitner P, Dustdar S (2009) An end-to-end approach for qos-aware service composition. In: Enterprise Distributed Object Computing Conference, 2009. EDOC'09. IEEE International. IEEE, Auckland. pp 151–160
97. Sakellariou R, Zhao H (2004) A low-cost rescheduling policy for efficient mapping of workflows on grid systems. Sci Program 12(4):253–262
98. Sakellariou R, Zhao H, Tsiakkouri E, Dikaiakos MD (2007) Scheduling Workflows with Budget Constraints (Gorlatch S, Danelutto M, eds.). Springer, Boston, MA, pp 189–202
99. Schwiegelshohn U, Yahyapour R (1999) Resource allocation and scheduling in metasystems. In: International Conference on High-Performance Computing and Networking. Springer, Amsterdam. pp 851–860
100. Selvarani S, Sadhasivam GS (2010) Improved cost-based algorithm for task scheduling in cloud computing. In: Computational Intelligence and Computing Research (iccic), 2010 Ieee International Conference On. IEEE, Coimbatore. pp 1–5
101. Senturk IF, Balakrishnan P, Abu-Doleh A, Kaya K, Malluhi Q, Çatalyürek ÜV (2016) A resource provisioning framework for bioinformatics applications in multi-cloud environments. Futur Gener Comput Syst. Elsevier. http://www.sciencedirect.com/science/article/pii/S0167739X16301911
102. Shah R, Veeravalli B, Misra M (2007) On the design of adaptive and decentralized load balancing algorithms with load estimation for computational grid environments. IEEE Trans Parallel Distrib Syst 18(12):1675–1686
103. Sharifi M, Shahrivari S, Salimi H (2013) Pasta: a power-aware solution to scheduling of precedence-constrained tasks on heterogeneous computing resources. Computing 95(1):67–88
104. Shi Z, Jeannot E, Dongarra JJ (2006) Robust task scheduling in non-deterministic heterogeneous computing systems. In: 2006 IEEE International Conference on Cluster Computing. IEEE, Barcelona. pp 1–10
105. Sih GC, Lee EA (1993) A compile-time scheduling heuristic for interconnection-constrained heterogeneous processor architectures. IEEE Trans Parallel Distrib Syst 4(2):175–187
106. Simao J, Veiga L (2013) Flexible slas in the cloud with a partial utility-driven scheduling architecture. In: Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference On. IEEE, Bristol Vol. 1. pp 274–281
107. Simarro JLL, Moreno-Vozmediano R, Montero RS, Llorente IM (2011) Dynamic placement of virtual machines for cost optimization in multi-cloud environments. In: High Performance Computing and Simulation (HPCS), 2011 International Conference On. IEEE, Istanbul. pp 1–7
108. Singh S, Chana I (2015) Cloud resource provisioning: survey, status and future research directions. Knowl Inf Syst 49(3):1005–69. https://link.springer.com/article/10.1007/s10115-016-0922-3
109. Singh S, Chana I (2016) A survey on resource scheduling in cloud computing: Issues and challenges. J Grid Comput 14(2):217–264
110. Slominski A, Muthusamy V, Khalaf R (2015) Building a multi-tenant cloud service from legacy code with docker containers. In: Cloud Engineering (IC2E), 2015 IEEE International Conference On. IEEE, Tempe. pp 394–396
111. Smanchat S, Viriyapant K (2015) Taxonomies of workflow scheduling problem and techniques in the cloud. Futur Gener Comput Syst 52:1–12
112. Sotiriadis S, Bessis N, Antonopoulos N (2011) Towards inter-cloud schedulers: A survey of meta-scheduling approaches. In: P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2011 International Conference On. IEEE, Barcelona. pp 59–66
113. Subramani V, Kettimuthu R, Srinivasan S, Sadayappan S (2002) Distributed job scheduling on computational grids using multiple simultaneous requests. In: High Performance Distributed Computing, 2002. HPDC-11 2002. Proceedings. 11th IEEE International Symposium On. IEEE, Edinburgh. pp 359–366
114. Talukder A, Kirley M, Buyya R (2009) Multiobjective differential evolution for scheduling workflow applications on global grids. Concurr Comput Pract Experience 21(13):1742–1756
115. Taylor IJ, Deelman E, Gannon DB, Shields M (2014) Workflows for e-Science: Scientific Workflows for Grids. Springer, London, UK
116. Tian F, Chen K (2011) Towards optimal resource provisioning for running mapreduce programs in public clouds. In: Cloud Computing (CLOUD), 2011 IEEE International Conference On. IEEE, Washington. pp 155–162
117. Topcuoglu H, Hariri S, Wu M-Y (2002) Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE Trans Parallel Distrib Syst 13(3):260–274
118. Tsai YL, Huang KC, Chang HY, Ko J, Wang ET, Hsu CH (2012) Scheduling multiple scientific and engineering workflows through task clustering and best-fit allocation. In: 2012 IEEE Eighth World Congress on Services. IEEE, Honolulu. pp 1–8
119. Varalakshmi P, Ramaswamy A, Balasubramanian A, Vijaykumar P (2011) An optimal workflow based scheduling and resource allocation in cloud. In: International Conference on Advances in Computing and Communications. Springer, Kochi. pp 411–420
120. Vecchiola C, Calheiros RN, Karunamoorthy D, Buyya R (2012) Deadline-driven provisioning of resources for scientific applications in hybrid clouds with aneka. Futur Gener Comput Syst 28(1):58–65
121. Venkatachalam V, Franz M (2005) Power reduction techniques for microprocessor systems. ACM Comput Surv (CSUR) 37(3):195–237
122. Wang CM, Chen HM, Hsu CC, Lee J (2010) Dynamic resource selection heuristics for a non-reserved bidding-based grid environment. Futur Gener Comput Syst 26(2):183–197
123. Wang L, Zhan J, Shi W, Liang Y (2012a) In cloud, can scientific communities benefit from the economies of scale? IEEE Trans Parallel Distrib Syst 23(2):296–303. doi:10.1109/TPDS.2011.144
124. Wang M, Ramamohanarao K, Chen J (2012b) Dependency-based risk evaluation for robust workflow scheduling. In: Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International. IEEE, Shanghai. pp 2328–2335
125. Weingärtner R, Bräscher GB, Westphall CB (2015) Cloud resource management: A survey on forecasting and profiling models. J Netw Comput Appl 47:99–106
126. Weissman JB, Grimshaw AS (1996) A federated model for scheduling in wide-area systems. In: High Performance Distributed Computing, 1996., Proceedings of 5th IEEE International Symposium On. IEEE, Syracuse. pp 542–550
127. Wu F, Wu Q, Tan Y (2015) Workflow scheduling in cloud: a survey. J Supercomput 71(9):3373–3418
128. Wu Z, Ni Z, Gu L, Liu X (2010) A revised discrete particle swarm optimization for cloud workflow scheduling. In: Computational Intelligence and Security (CIS), 2010 International Conference On. IEEE, Nanning. pp 184–188
129. Wu Z, Liu X, Ni Z, Yuan D, Yang Y (2013) A market-oriented hierarchical scheduling strategy in cloud workflow systems. J Supercomput 63(1):256–293
130. Xiao P, Hu ZG, Zhang YP (2013) An energy-aware heuristic scheduling for data-intensive workflows in virtualized datacenters. J Comput Sci Technol 28(6):948–961
131. Xiao Y, Lin C, Jiang Y, Chu X, Shen X (2010) Reputation-based qos provisioning in cloud computing via dirichlet multinomial model. In:

Communications (ICC), 2010 IEEE International Conference On. IEEE, China. pp 1–5

132. Xu B, Zhao C, Hu E, Hu B (2011) Job scheduling algorithm based on berger model in cloud environment. Adv Eng Softw 42(7):419–425
133. Xu M, Cui L, Wang H, Bi Y (2009) A multiple qos constrained scheduling strategy of multiple workflows for cloud computing. In: 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications. IEEE, Chengdu. pp 629–634
134. Yassa S, Chelouah R, Kadima H, Granado B (2013) Multi-objective approach for energy-aware workflow scheduling in cloud computing environments. Sci World J 2013:e350934. https://www.hindawi.com/journals/tswj/2013/350934/abs/, doi:10.1155/2013/350934
135. Yi S, Andrzejak A, Kondo D (2012) Monetary cost-aware checkpointing and migration on amazon cloud spot instances. IEEE Trans Serv Comput 5(4):512–524
136. Yoo S, Kim S (2013) Sla-aware adaptive provisioning method for hybrid workload application on cloud computing platform. In: Proceedings of the International Multiconference of Engineers and Computer Scientists, Hong Kong Vol. 1
137. Yu J, Buyya R (2006) Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms. Sci Program 14(3-4):217–230
138. Yu J, Buyya R, Tham CK (2005) Cost-based scheduling of scientific workflow applications on utility grids. In: First International Conference on e-Science and Grid Computing (e-Science'05). IEEE, Melbourne. p 8
139. Yu J, Kirley M, Buyya R (2007) Multi-objective planning for workflow execution on grids. In: Proceedings of the 8th IEEE/ACM International Conference on Grid Computing. IEEE Computer Society, Austin. pp 10–17
140. Yu J, Ramamohanarao K, Buyya R (2009a) Deadline/budget-based scheduling of workflows on utility grids. Market-Oriented Grid Util Comput 200(9):427–450
141. Yu J, Ramamohanarao K, Buyya R (2009b) Deadline/budget-based scheduling of workflows on utility grids. Market-Oriented Grid Util Comput 200(9):427–450
142. Yu Z, Shi W (2008a) A planner-guided scheduling strategy for multiple workflow applications. In: 2008 International Conference on Parallel Processing-Workshops. IEEE, Portland. pp 1–8
143. Zaman S, Grosu D Combinatorial auction-based dynamic vm provisioning and allocation in clouds. In: Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference On. IEEE, Athens. pp 107–114
144. Zeng L, Veeravalli B, Li X (2012) Scalestar: Budget conscious scheduling precedence-constrained many-task workflow applications in cloud. In: 2012 IEEE 26th International Conference on Advanced Information Networking and Applications. IEEE, Fukuoka. pp 534–541
145. Zhang J, Yousif M, Carpenter R, Figueiredo RJ (2007a) Application resource demand phase analysis and prediction in support of dynamic resource provisioning. In: Fourth International Conference on Autonomic Computing (ICAC'07). IEEE, Jacksonville. pp 12–12
146. Zhang J, Kim J, Yousif M, Carpenter R, et al. (2007b) System-level performance phase characterization for on-demand resource provisioning. In: 2007 IEEE International Conference on Cluster Computing. IEEE, Austin. pp 434–439
147. Zhang Q, Cheng L, Boutaba R Cloud computing: state-of-the-art and research challenges. J Internet Serv Appl 1(1):7–18
148. Zhang Q, Zhani MF, Zhang S, Zhu Q, Boutaba R, Hellerstein JL (2012) Dynamic energy-aware capacity provisioning for cloud computing environments. In: Proceedings of the 9th International Conference on Autonomic Computing. ACM, London. pp 145–154
149. Zhao H, Sakellariou R (2006) Scheduling multiple dags onto heterogeneous systems. In: Proceedings 20th IEEE International Parallel & Distributed Processing Symposium. IEEE, Rhodes Island. p 14
150. Zhao Y, Li Y, Raicu I, Lu S, Tian W, Liu H (2015) Enabling scalable scientific workflow management in the cloud. Futur Gener Comput Syst 46:3–16
151. Zheng W, Sakellariou R (2011) Budget-deadline constrained workflow planning for admission control in market-oriented environments. In: International Workshop on Grid Economics and Business Models. Springer. pp 105–119
152. Zheng W, Sakellariou R (2013) Stochastic dag scheduling using a monte carlo approach. J Parallel Distrib Comput 73(12):1673–1689
153. Zhong H, Tao K, Zhang X (2010) An approach to optimized resource scheduling algorithm for open-source cloud systems. In: 2010 Fifth Annual ChinaGrid Conference. IEEE, Guangzhou. pp 124–129
154. Zhou AC, He B, Liu C (2016) Monetary cost optimizations for hosting workflow-as-a-service in iaas clouds. IEEE Trans Cloud Comput 4(1):34–48